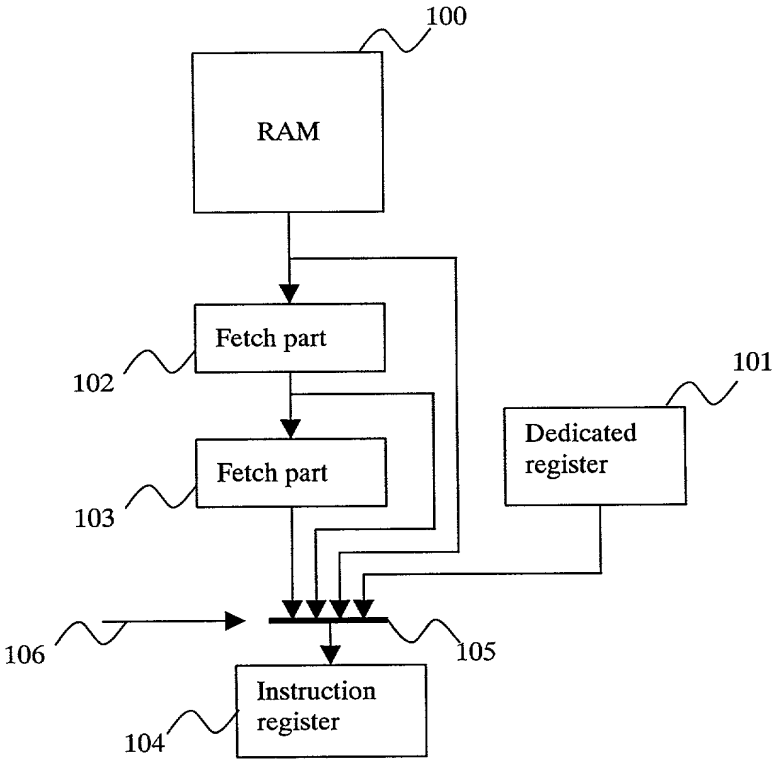
100
RAM
Fetch part
102
Fetch part
103
101
Dedicated register
106
105
Instruction register
104

Fig. 1

Branch is approved

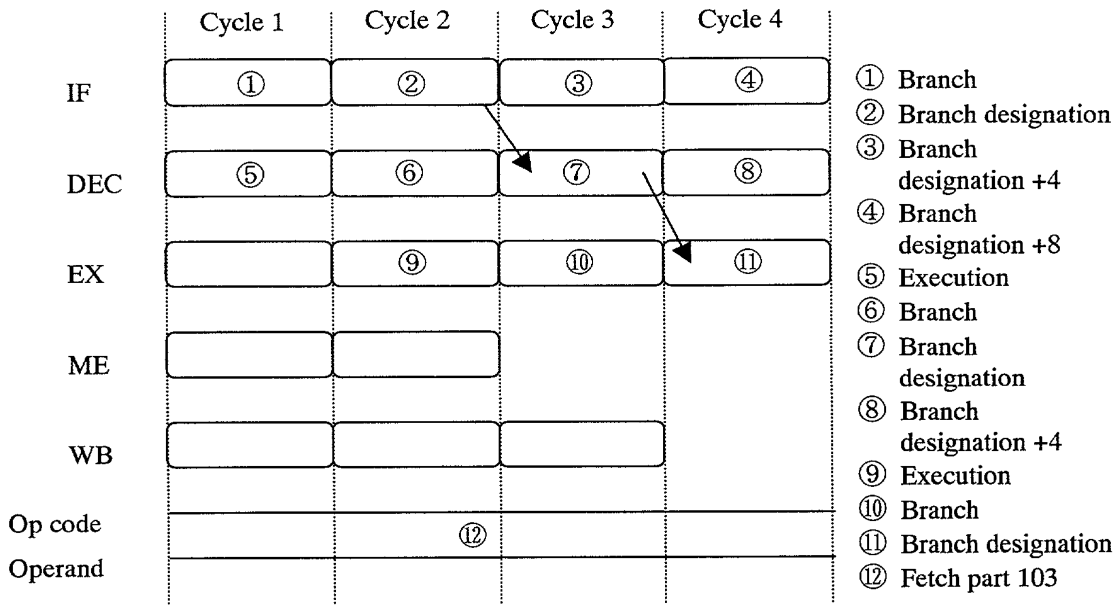

Fig. 2 A

Branch is not approved

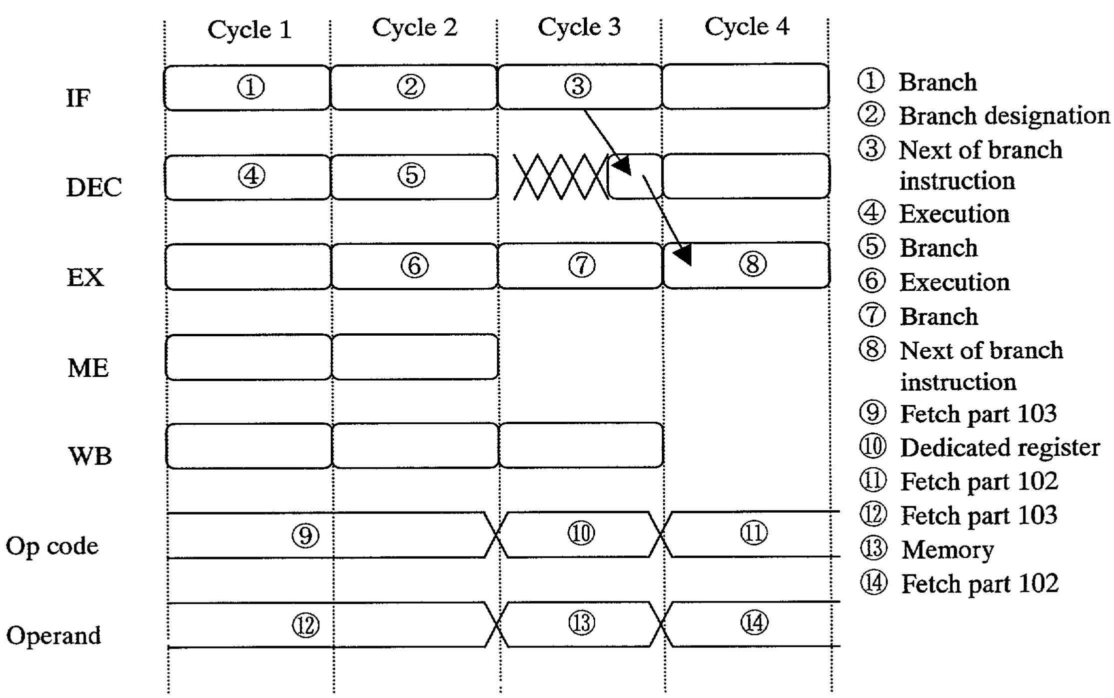

Fig. 2 B

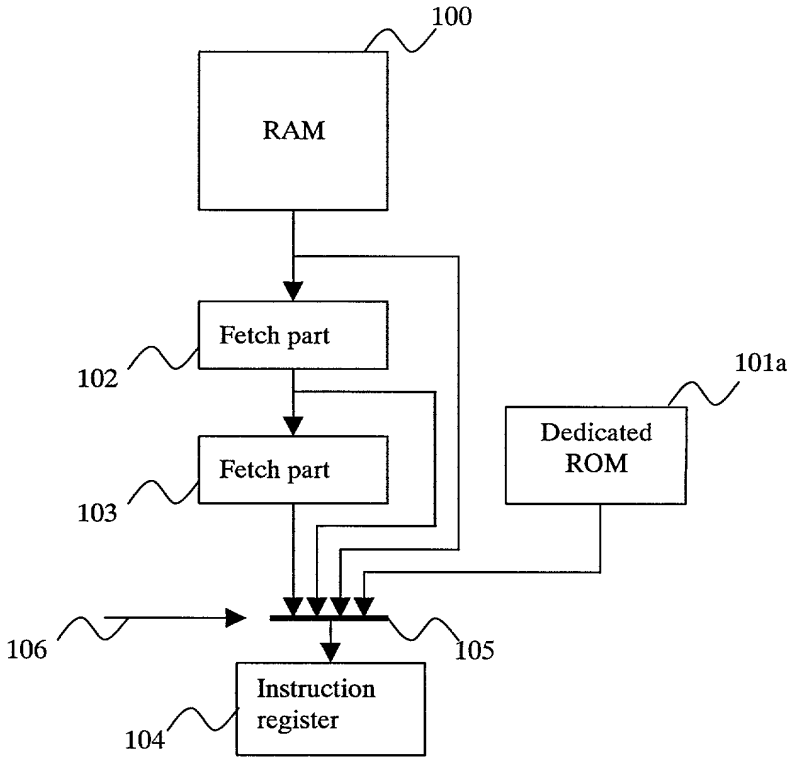
100
RAM
Fetch part
102
Fetch part
103
101a
Dedicated ROM
106
105
Instruction register
104

Fig. 3

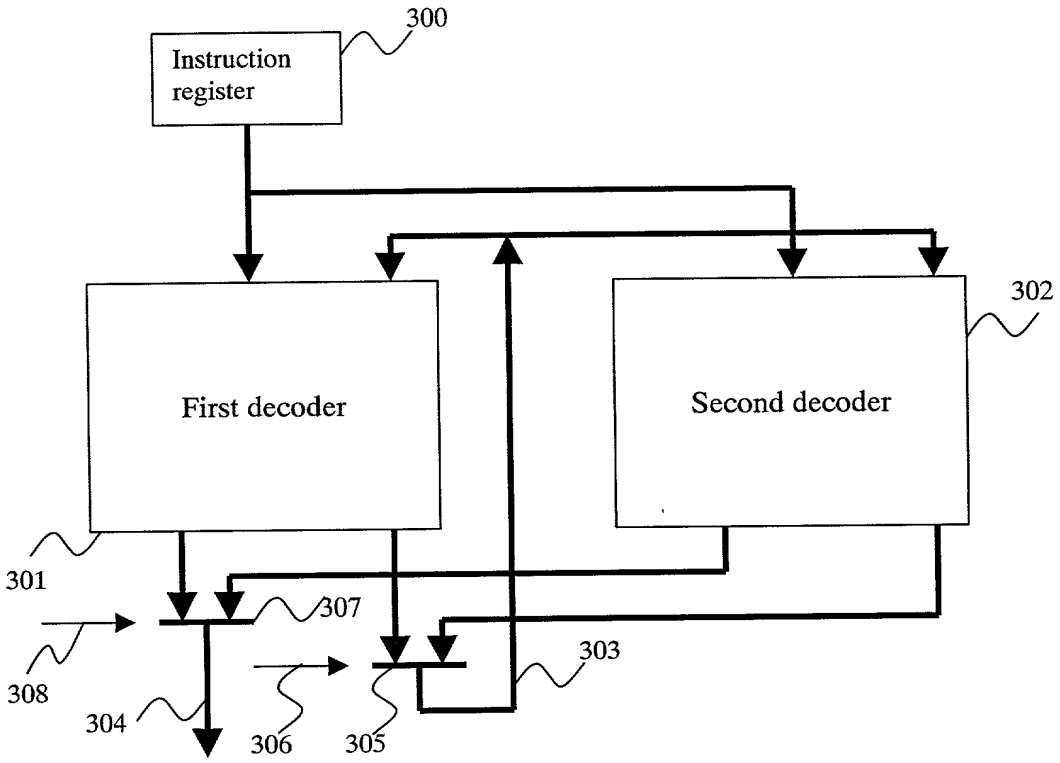
300
Instruction register
First decoder
Second decoder
302
301
307
308
303
304
306
305

Fig. 4

Branch is approved

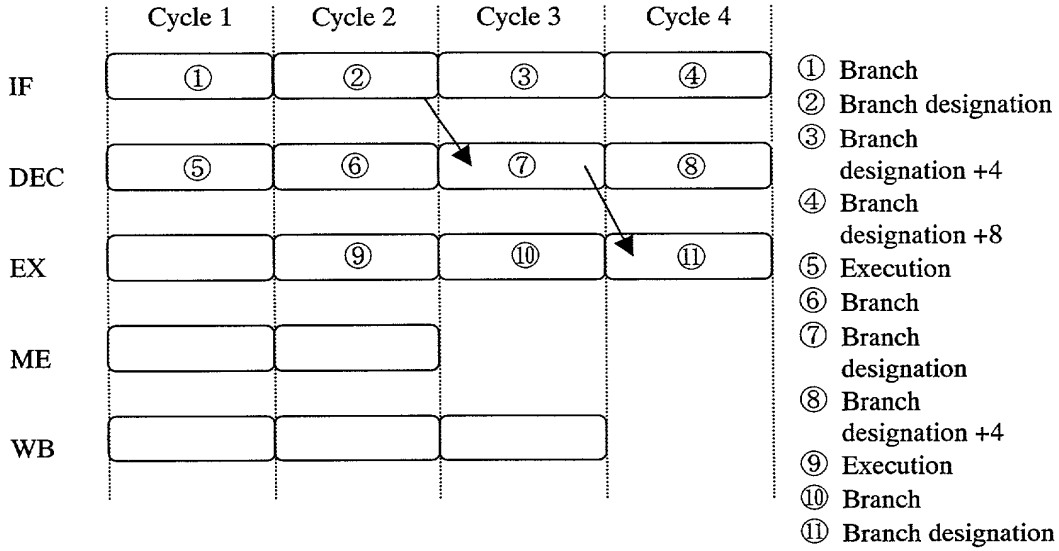

Fig. 5 A

Branch is not approved

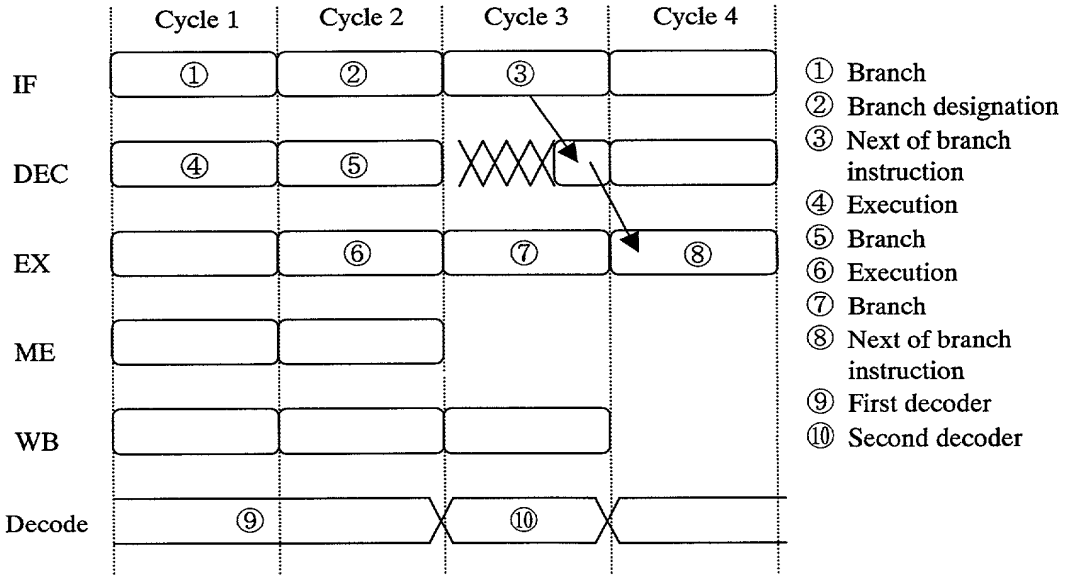

Fig. 5 B

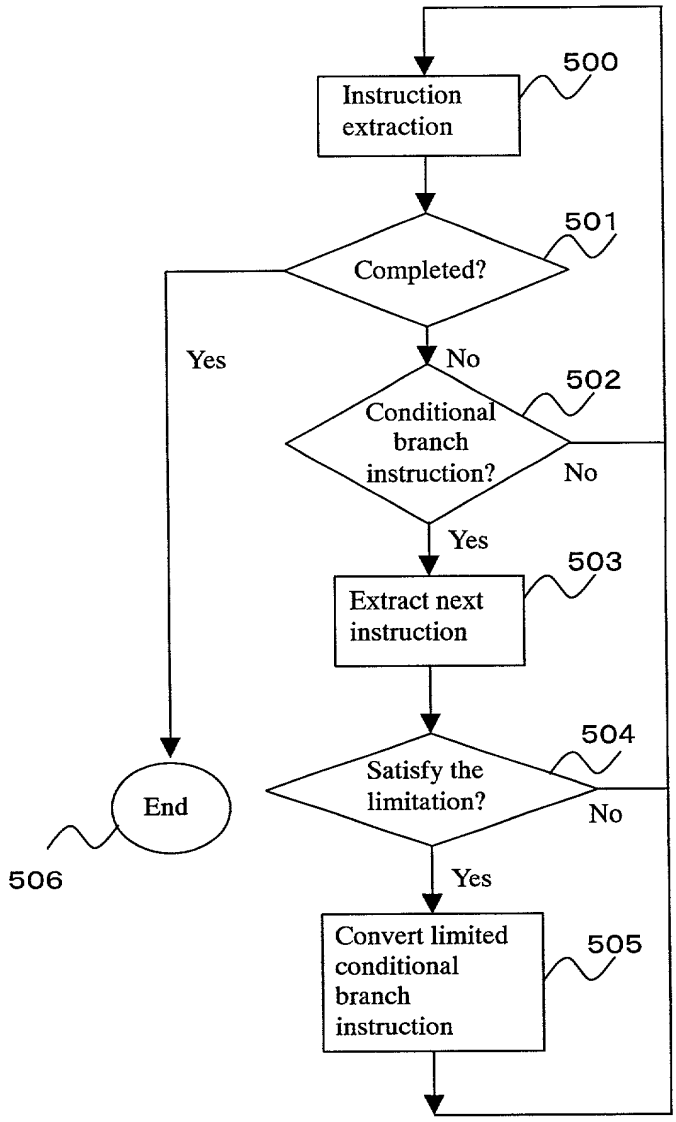
Instruction extraction
500
Completed?
501
Yes
No
Conditional branch instruction?
502
No
Yes
Extract next instruction
503
Satisfy the limitation?
504
No
End
506
Yes
Convert limited conditional branch instruction
505

Fig. 6

Example of case statement	Assembler description	Converted result of instruction sequence
switch(mode)	ld d0,(mode)	ld d0,(mode)
case: 1 a=b+1;	cmp d0,1	cmp d0,1
case: 3 a=c;	jz	cjz
case: 5 a=b+c;	cmp d0,3	cmp d0,3
:	jz	cjz
:	cmp d0,5	cmp d0,5
	jz	cjz
	:	:
	:	:

Fig. 7A Fig. 7B Fig. 7C

Branch is approved

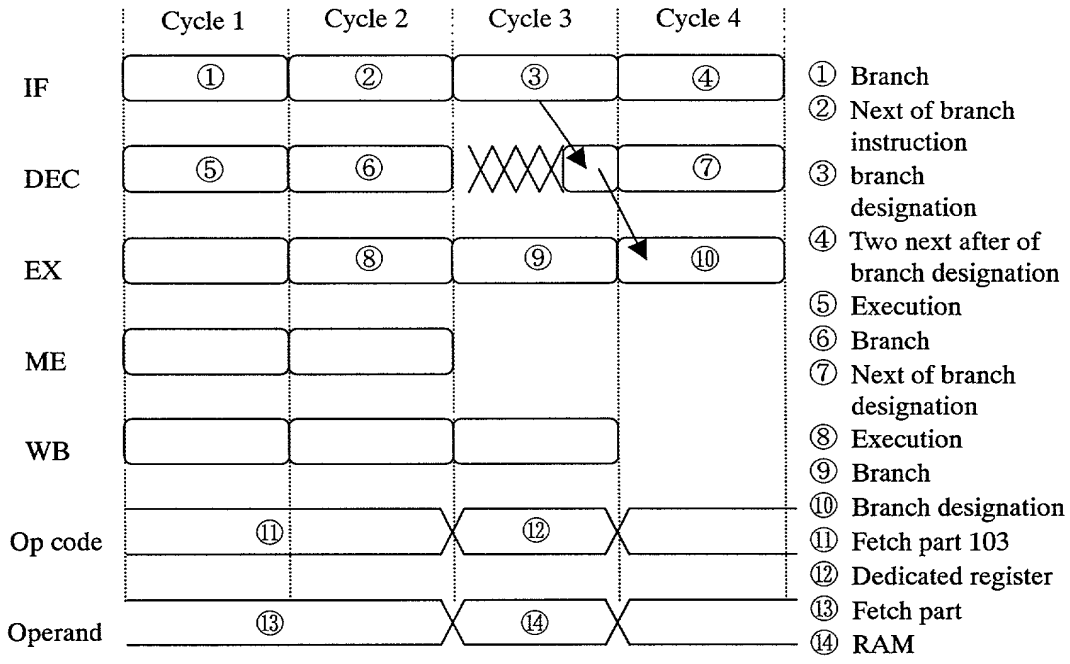

Fig. 8 A

Branch is not approved

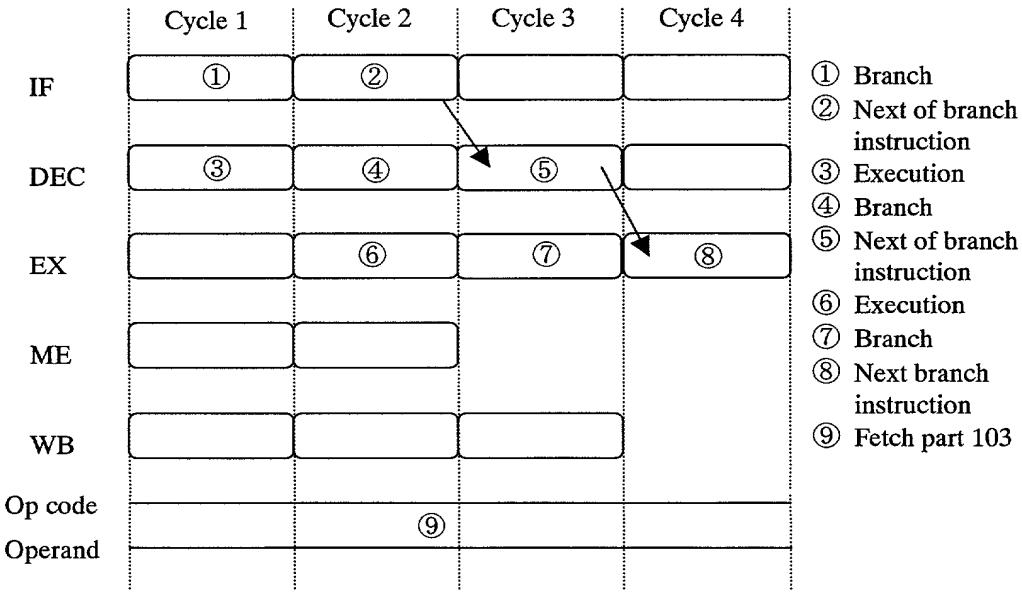

Fig. 8 B

Branch is approved

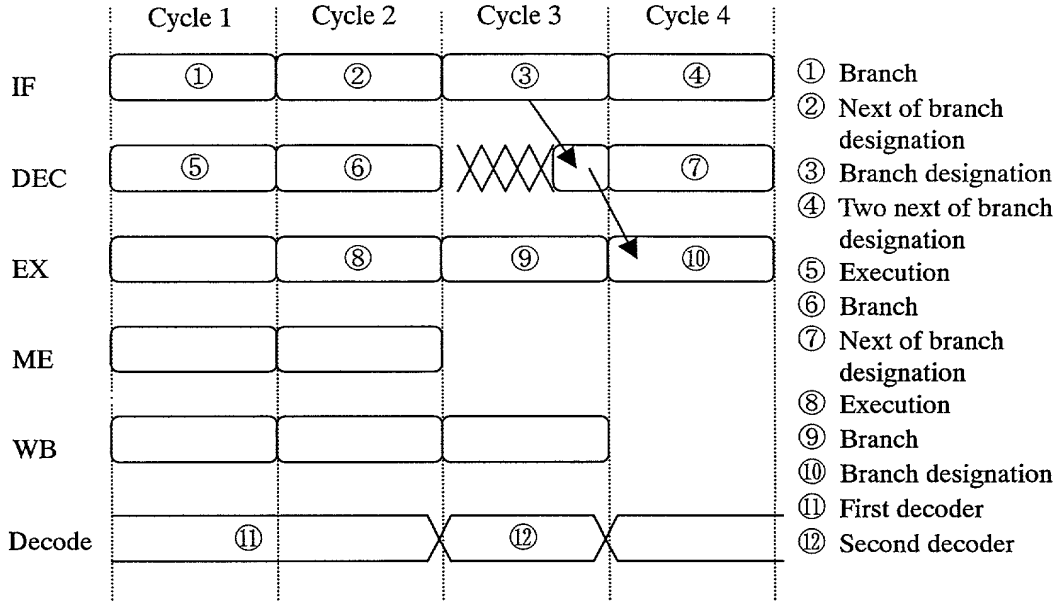

Fig. 9 A

Branch is not approved

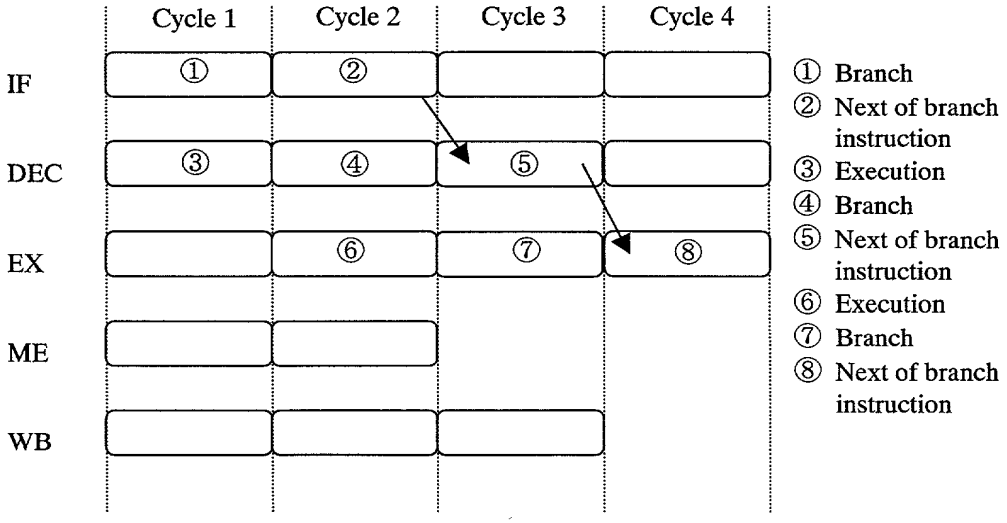

Fig. 9 B

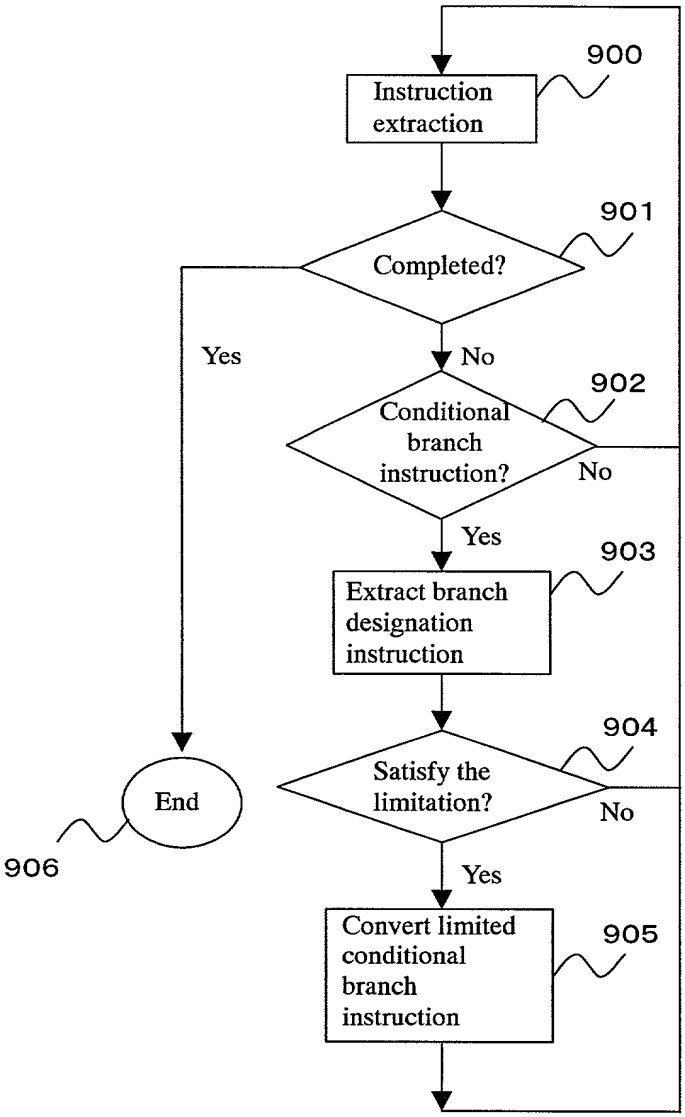
900
Instruction extraction
901
Completed?
Yes
No
902
Conditional branch instruction?
No
Yes
903
Extract branch designation instruction
904
Satisfy the limitation?
No
Yes
End
906
Convert limited conditional branch instruction
905

Fig. 10

Example of case statement

```
switch(mode)
case: 1 a=b+1;
case: 3 a=c;
case: 5 a=b+c;
  :
  :
```

Fig. 11A

Assembler description

```
    ld d0,(mode)
    cmp d0,1
    jnz A
    ld d0,(b)
    add d0,1
    st d0,(a)
    jmp END
A:  cmp d0,3
    jnz B
    ld d0,(c)
    st d0,(a)
    jmp END
B:  cmp d0,5
    jnz C
    ld d0,(c)
    ld d1,(b)
    add d0,d1
    st d0,(a)
    :
    :
END:
```

Fig. 11B

Converted result of instruction sequence

```
    ld d0,(mode)
    cmp d0,1
    cjnz A
    ld d0,(b)
    add d0,1
    st d0,(a)
    jmp END
A:  cmp d0,3
    cjnz B
    ld d0,(c)
    st d0,(a)
    jmp END
B:  cmp d0,5
    cjnz C
    ld d0,(c)
    ld d1,(b)
    add d0,d1
    st d0,(a)
    :
    :
END:
```

Fig. 11C

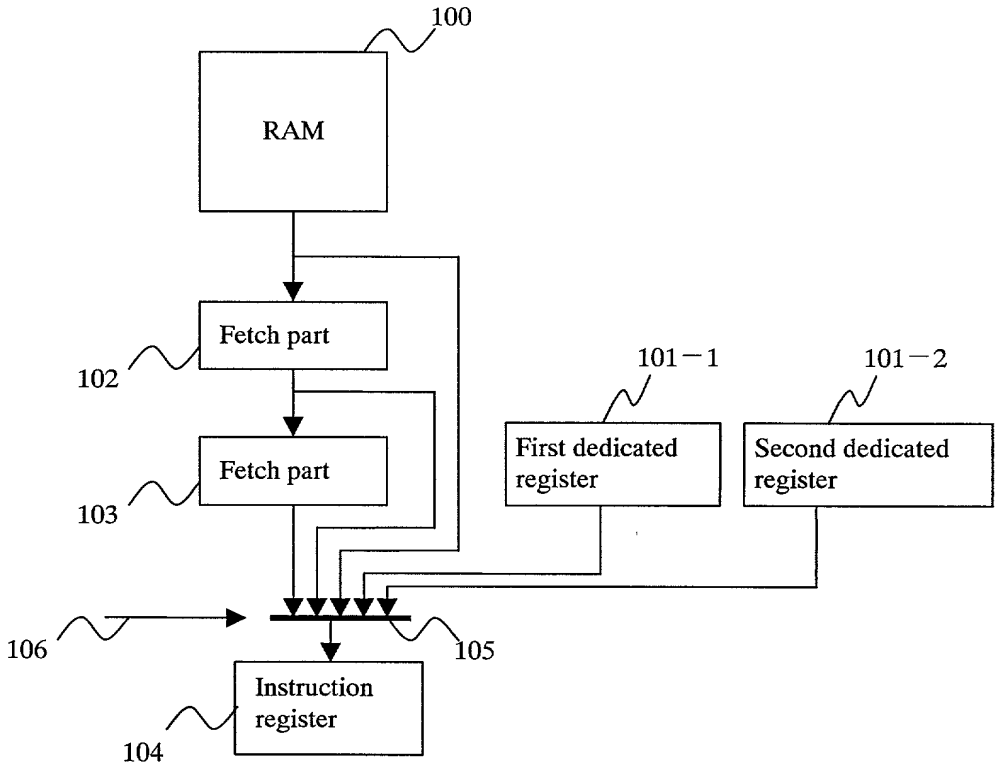
100
RAM
Fetch part
102
Fetch part
103
101−1
First dedicated register
101−2
Second dedicated register
106
105
Instruction register
104

Fig. 12

Unconditional branch

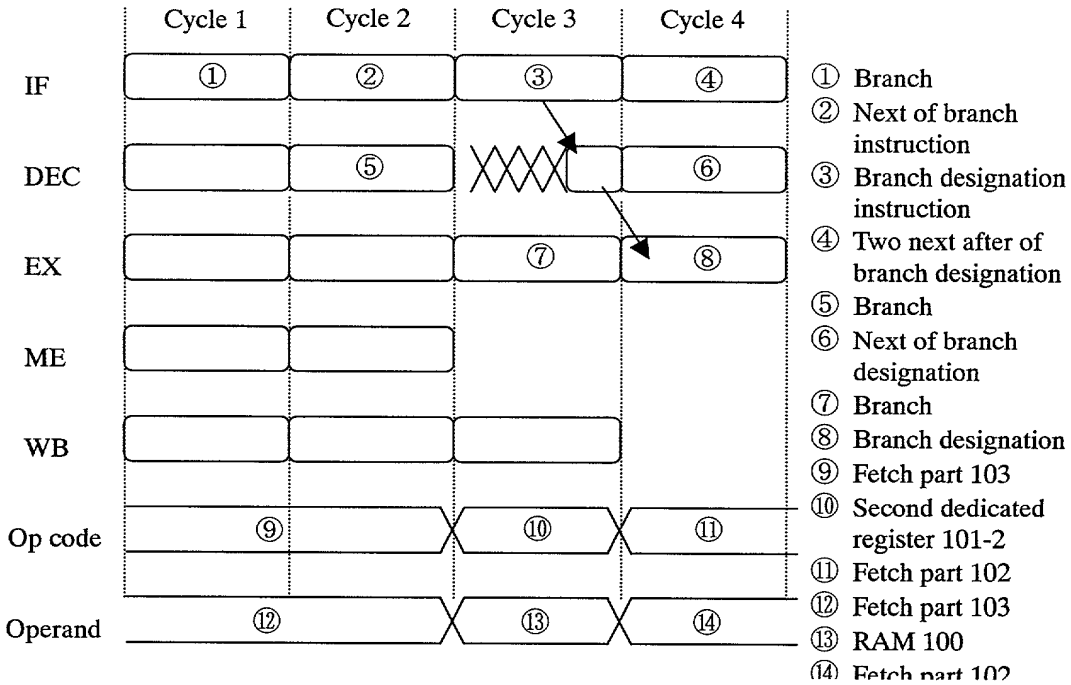

Fig. 13

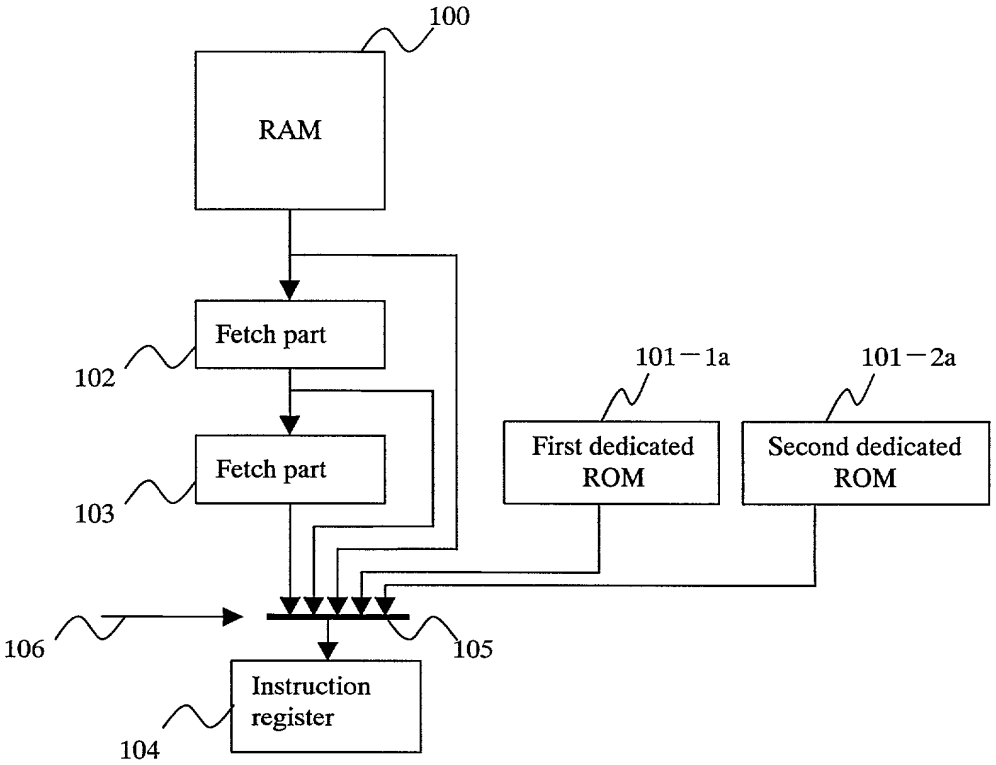
100
RAM
Fetch part
102
Fetch part
103
101−1a
First dedicated ROM
101−2a
Second dedicated ROM
106
105
Instruction register
104

Fig. 14

Unconditional branch

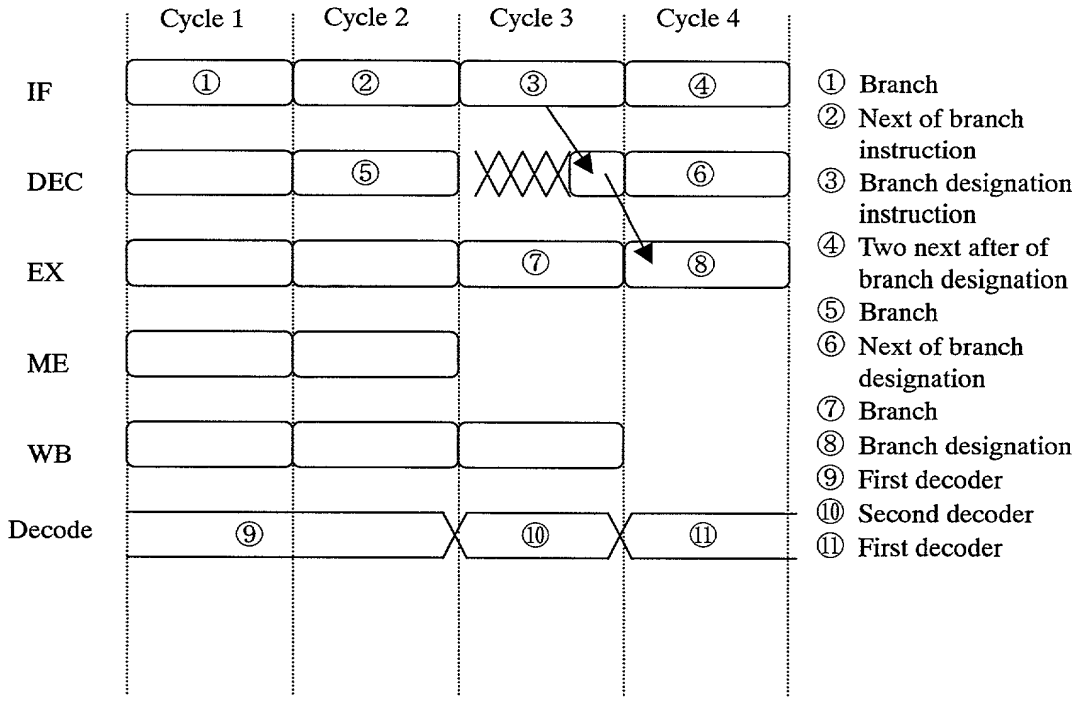

Fig. 15

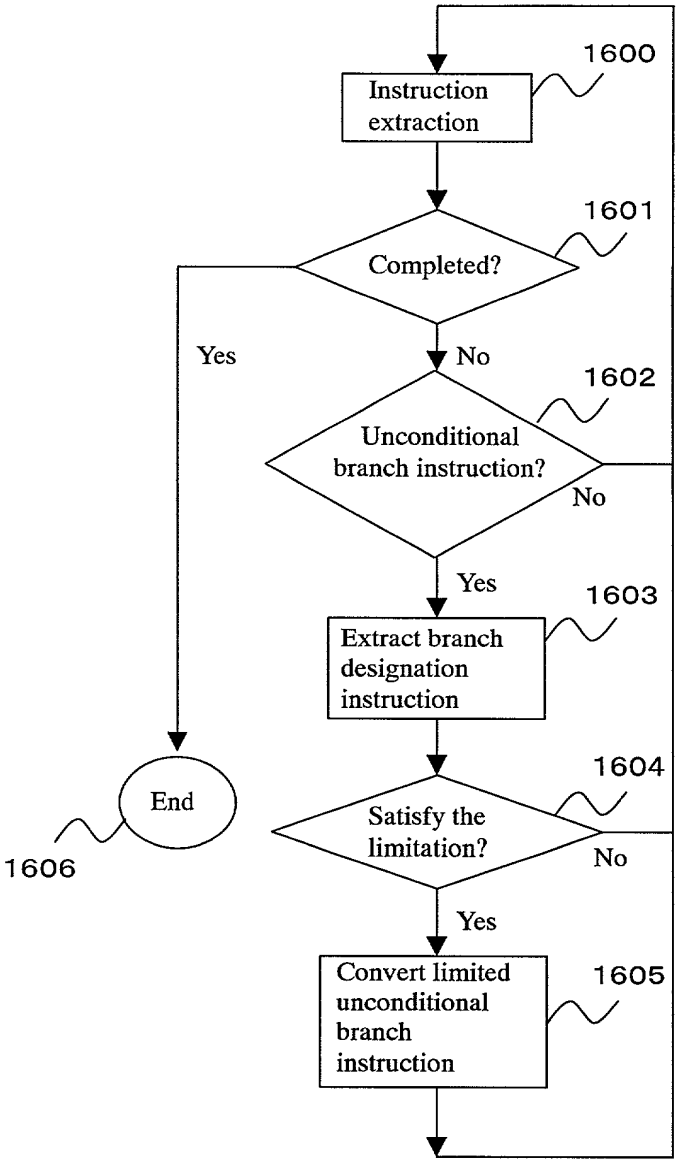
1600
Instruction extraction
1601
Completed?
Yes
No
1602
Unconditional branch instruction?
No
Yes
1603
Extract branch designation instruction
1604
Satisfy the limitation?
No
Yes
1605
Convert limited unconditional branch instruction
End
1606

Fig. 16

Example of case statement

```
switch(mode1){
    case: 1 a1=b1+1;
    case: 3 a1=c1;
    case: 5 a1=b1+c1;
    }
switch(mode2){
    case: 1 a2=b2+1;
    case: 3 a2=c2;
    case: 5 a2=b2+c2;
    }
```

Fig. 17

Assembler description

```
      ld d0,(mode1)
      cmp d0,1
      jnz A1
      ld d0,(b)
      add d0,1
      st d0,(a)
      jmp C1
A1:   cmp d0,3
      jnz B1
      ld d0,(c)
      st d0,(a)
      jmp C1
B1:   cmp d0,5
      jnz C1
      ld d0,(c)
      ld d1,(b)
      add d0,d1
      st d0,(a)
```

```
C1    ld d0,(mode2)
      cmp d0,1
      jnz A2
      ld d0,(b)
      add d0,1
      st d0,(a)
      jmp C2
A1:   cmp d0,3
      jnz B2
      ld d0,(c)
      st d0,(a)
      jmp C2
B1:   cmp d0,5
      jnz C2
      ld d0,(c)
      ld d1,(b)
      add d0,d1
      st d0,(a)
C2:   • • •
      • • •
```

Fig. 18

Instruction sequence after conversion

```
     ld d0,(mode1)
     cmp d0,1
     cjnz A1
     ld d0,(b)
     add d0,1
     st d0,(a)
     cjmp C1
A1:  cmp d0,3
     cjnz B1
     ld d0,(c)
     st d0,(a)
     cjmp C1
B1:  cmp d0,5
     cjnz C1
     ld d0,(c)
     ld d1,(b)
     add d0,d1
     st d0,(a)
```

```
C1   ld d0,(mode2)
     cmp d0,1
     cjnz A2
     ld d0,(b)
     add d0,1
     st d0,(a)
     cjmp C2
A1:  cmp d0,3
     cjnz B2
     ld d0,(c)
     st d0,(a)
     cjmp C2
B1:  cmp d0,5
     cjnz C2
     ld d0,(c)
     ld d1,(b)
     add d0,d1
     st d0,(a)
C2:  • • •
     • • •
```

Fig. 19

Branch is approved

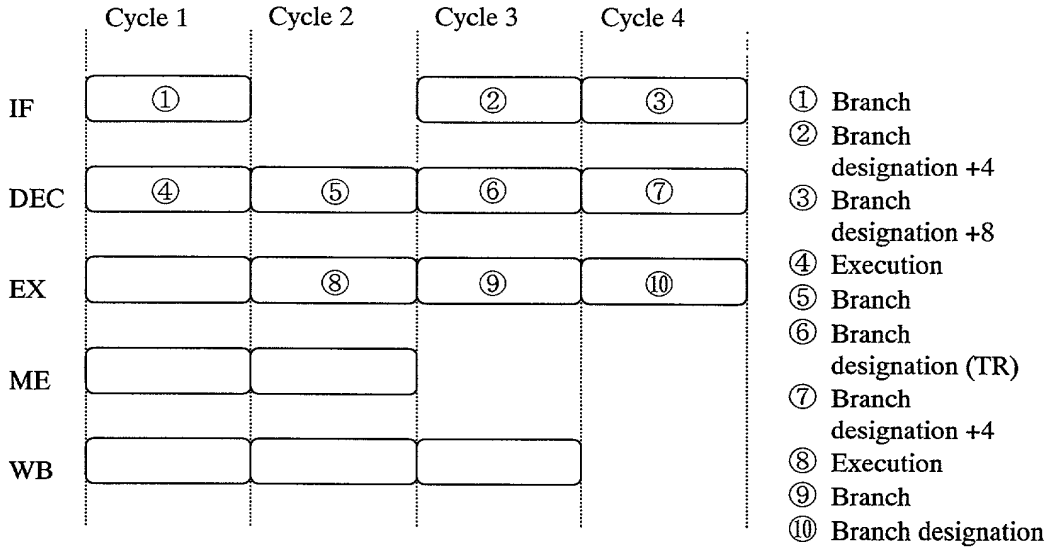

Fig. 20 A PRIOR ART

Branch is not approved

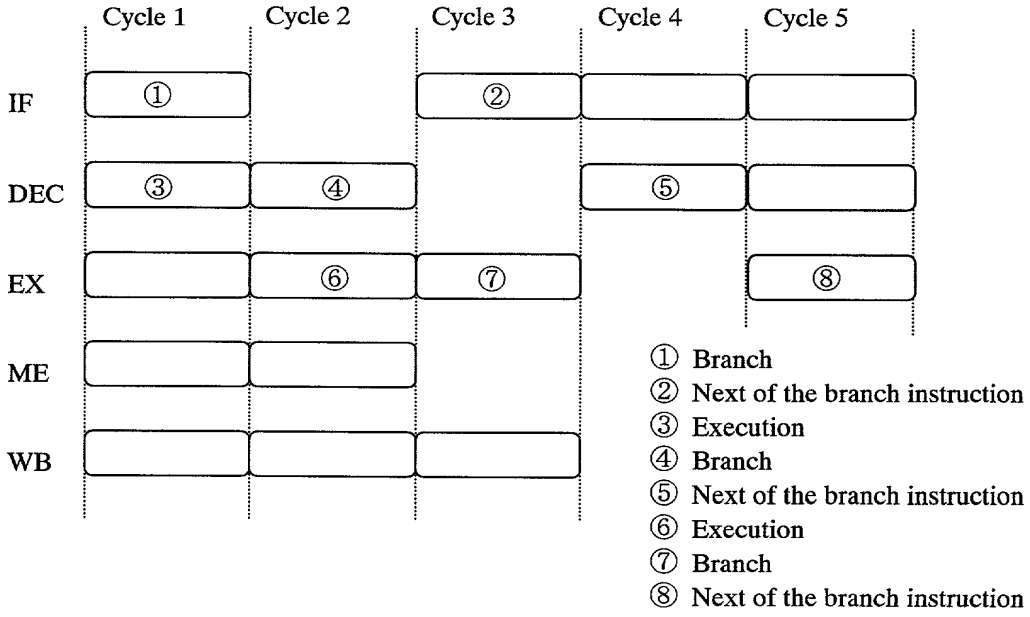

Fig. 20 B PRIOR ART

Branch is approved

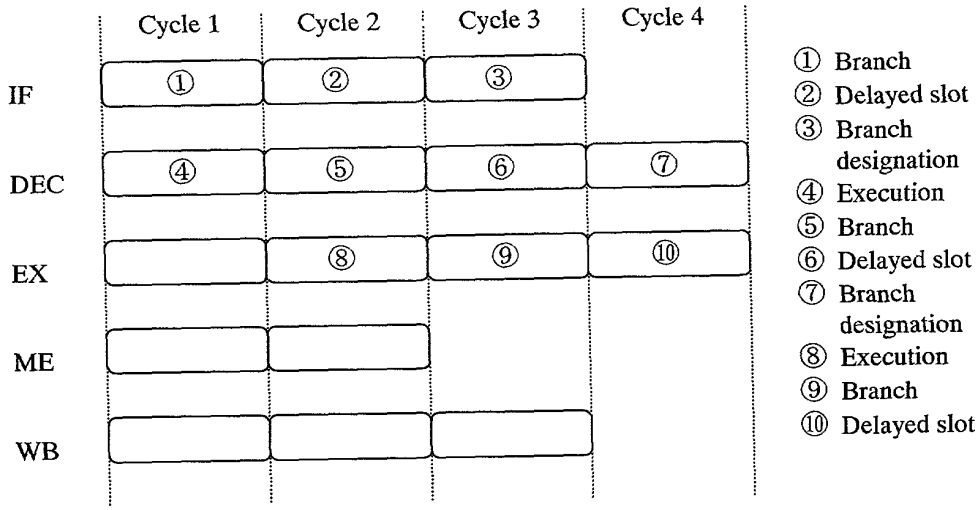

Fig. 21 A PRIOR ART

Branch is not approved

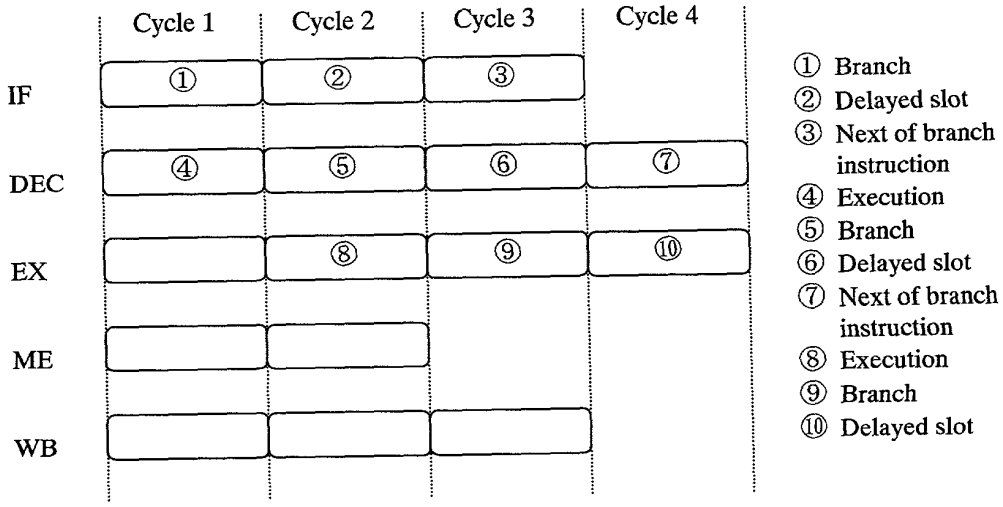

Fig. 21 B PRIOR ART

Example of case statement

```
switch(mode)
case: 1 a=b+1;
case: 3 a=c;
case: 5 a=b+c;
  :
  :
```

Fig. 22A

PRIOR ART

Assembler description

```
ld d0,(mode)
cmp d0,1
jz
cmp d0,3
jz
cmp d0,5
jz
:
:
```

Fig. 22B

PRIOR ART